Towards a Hybrid Recommendation System for a Sound Library

Jason Smith
jsmith775@gatech.edu
Center for Music Technology
Georgia Institute of Technology
Atlanta, GA

Dillon Weeks
dweeks7@gatech.edu
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA

Mikhail Jacob
mikhail.jacob@gatech.edu
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA

Jason Freeman
jason.freeman@gatech.edu
Center for Music Technology
Georgia Institute of Technology
Atlanta, GA

Brian Magerko
magerko@gatech.edu
School of Literature, Media, and Communication
Georgia Institute of Technology
Atlanta, GA

ABSTRACT

Recommendation systems are widespread in music distribution and discovery services but far less common in music production software such as EarSketch, an online learning environment that engages learners in writing code to create music. The EarSketch interface contains a sound library that learners can access through a browser pane. The current implementation of the sound browser includes basic search and filtering functionality but no mechanism for sound discovery, such as a recommendation system. As a result, users have historically selected a small subsection of sounds in high frequencies, leading to lower compositional diversity. In this paper, we propose a recommendation system for the EarSketch sound browser which uses collaborative filtering and audio features to suggest sounds.

CCS CONCEPTS

• **Human-centered computing** → **User interface design**; • **Applied computing** → **Sound and music computing**.

KEYWORDS

recommendation systems, interface design, music

ACM Reference Format:
Jason Smith, Dillon Weeks, Mikhail Jacob, Jason Freeman, and Brian Magerko. 2019. Towards a Hybrid Recommendation System for a Sound Library. In *Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019* , 6 pages.

1 INTRODUCTION

EarSketch [7] is an online environment for learning computer programming and audio loop-based music composition. Students write JavaScript or Python scripts to algorithmically generate musical compositions. The user interface borrows design cues from both integrated development environments (IDEs) and digital audio workstation (DAW) software, combining a code editor and console with a multi-track audio timeline and sound browser. EarSketch has primarily been used in high school and college computer science classrooms, with over 300,000 users to date [5].

In previous research in EarSketch classrooms, significant relationships have been found between student perceptions of authenticity – including their desire to share personally expressive work with others – and student attitudes towards computing [9]. Exploration of a larger number of musical ideas – including the sounds that form the building blocks of student compositions in EarSketch – may magnify a student's capacity to create personally expressive compositions.

EarSketch contains a library of over 3,500 sounds for students to use in their compositions. The sounds were created by musicians Richard Devine and Young Guru specifically for EarSketch and consist of multi-measure audio loops that are separated by instrument and span over 20 popular musical genres. However, a statistical analysis of scripts written by users showed that the vast majority of user projects used only a small subset of the sound library. Feedback from EarSketch users (found in the interviews section) showed that their lack of exploration was primarily the result of the difficulty in finding sounds that appealed to them. We propose, therefore, that providing users with an easier mechanism for exploring the sound library will enable them to find and use audio loops that spur further musical creativity and personal expression, while ultimately furthering their learning about music and coding through EarSketch.

We have explored the addition of a recommendation (or recommender) system, after conducting user studies, as a method of encouraging users to explore more of the EarSketch sound library in their scripts. Recommendation systems are widespread in music distribution and discovery platforms (where they operate at the song level) but far less common in music production workflows (where they could operate at the sound clip level). Recommendation

IUI Workshops'19, March 20, 2019, Los Angeles, USA

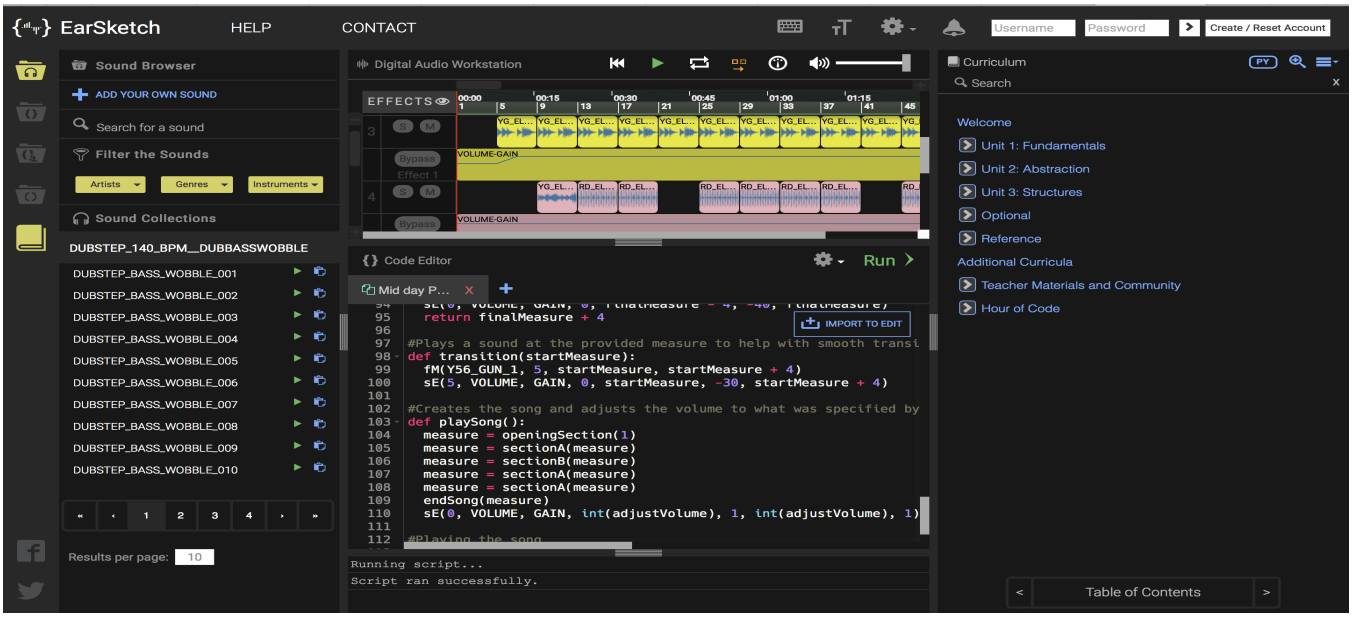

Figure 1: View of EarSketch browser interface.

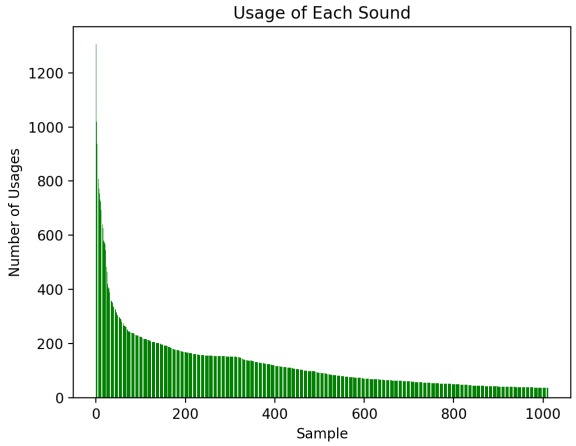

Figure 2: The EarSketch sounds used the highest number of times in 20,000 user scripts (highest 1,000 shown for legibility), showing under-utilization of the majority of the library.

systems suggest content to users that is most likely to appeal to them based on profiles of their preferences as well as content that they would most likely find novel, diverse, and unexpectedly useful (serendipitous) [1]. EarSketch could use such a recommendation system to automatically search through its sound library to find relevant sounds that encourage the user to explore novel, diverse, and serendipitous regions of the sound library.

Recommendation generation techniques include collaborative filtering, content-based filtering, and hybrid techniques. Collaborative filtering [1] involves comparing the current user to previous users in order to generate recommendations from what similar users in the past selected (for example [13]). Content-based filtering compares inherent properties of content to recommend items, such as with the use of audio feature-based deep learning [3] and calculation of short sample similarity metrics [14]. We can use a hybrid approach that combines both techniques to generate recommendations.

Some previous recommendation systems for sounds employed the Freesound sample library [4]. These projects used feature similarity calculations without co-usage statistics [12] or textual metadata to augment recommendations [11]. The proposed system for EarSketch differs from these examples by combining only audio similarity and co-usage to generate recommendations, reserving genre labels for manual user filtering.

In this article, we present our initial research on a recommendation system for discovering new sounds for use in EarSketch. The main contributions discussed are:

- An initial user-centered design process for systematically understanding how best to add an audio loop recommendation system into the EarSketch environment, including, the way users currently use the sound browser, the challenges to using it successfully, the kinds of recommendations users desire, and the best way to present users with recommendations.
- The initial application of a hybrid (collaborative and content-based filtering) recommendation system for sounds in a digital audio workstation, in contrast to song recommendation systems. This is a first step towards improving user exploration of the EarSketch sound library according to the user requirements and design principles arising from the initial user-centered design process.

- A proposed methodology for evaluating both the success of the recommendation system in providing users with relevant, novel, diverse, and serendipitous recommendations [1] and the relative importance of the different factors used to generate recommendations, as well as the usability of the sound browser with the recommendation system added.

The remainder of the paper describes the details of the user-centered design process for adding a recommendation system to the EarSketch sound browser and the initial prototype of the hybrid recommendation system resulting from that design process. The paper concludes by discussing the planned evaluation methodology, limitations of the current prototype, and future work.

2 USER RESEARCH AND INTERFACE DESIGN

An initial user study was conducted in order to gain a systematic understanding of how best to add a recommendation system into the EarSketch sound browser. This included understanding the different ways that users used the sound browser, the challenges they faced to use it successfully, the kinds of recommendations users desired, and the best ways to present recommendations to users. The study resulted in a set of requirements for the recommendation system and a redesign of the sound browser interface integrating the generated recommendations.

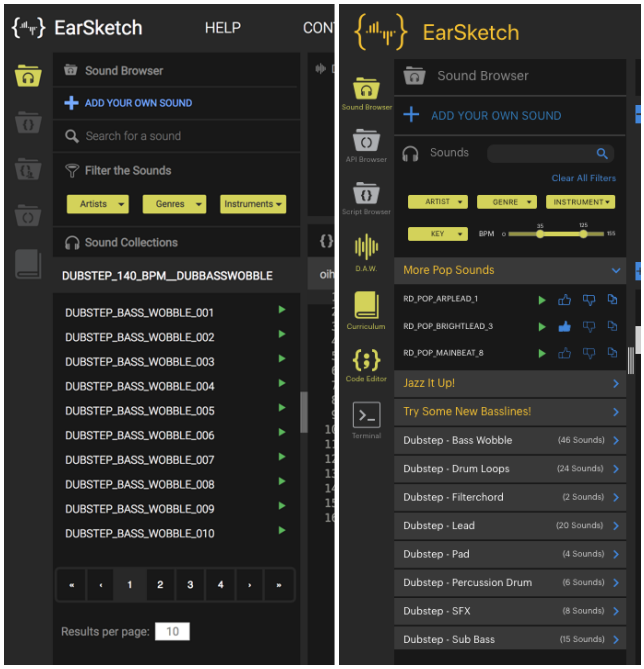

Figure 3: Original sound browser design prior to research activities (left) and sound browser design after research activities (right). Includes a like/dislike functionality, collapsible sound folders, new recommended sound folders with gold text to distinguish them as recommendations, and the addition of Key and BPM filters.

2.1 Initial Design

The sound browser experience prior to the addition of a recommendation system included sound folders that consisted of a title and a list of sounds corresponding to that title. For example, the sound folder titled "DUBSTEP 140 BPM DUBBASS WOBBLE" included a list of "DUBSTEP BASS WOBBLE" sounds underneath it, followed by other sound folders and their associated sounds. This list was navigated via scrolling and sounds were distributed across multiple pages within the browser. The user had the ability to favorite and preview these sounds from within the browser as well. The user also had the ability to discover sounds in the library via text search from the search bar along with the functionality to filter these sounds by artists, instruments, and genre.

2.2 Interviews and Survey of EarSketch Students

Four qualitative interviews were conducted with undergraduate students in an introductory programming course at a four-year college to explore current EarSketch users' challenges, behaviors, and interactions with the sound browser. This was done to identify the best opportunities for the recommendation system to fit their needs. These interviews were utilized to gather qualitative data such as reported behaviors, motivations behind those behaviors, opportunities for future designs and a recommendation integration. A quantitative survey was sent out to the same undergraduate class and received 55 responses. The survey was used to determine the prevalence of identified behaviors and preferences.

Participants reported being more inclined to use the *Instrument* and *Genre* filters than the *Artist* filter. In addition, users expressed their desire for a *Key* and *Beats-Per-Minute (BPM)* filter. This suggested the need to prioritize recommendations based on instruments, genres, keys, and BPM in the future.

Users reported that it was hard to discover groups of sounds they considered to be a good recommendation. They considered strong recommendations to be sounds that they liked that also fit in their script (relevant) and that they had not heard before (novel) or were not expecting (serendipitous). Discovering sounds similar to previously used sounds was of lesser importance to them. This confirmed that those users desired recommendations that were in accordance with the recommendation system goals defined by [1].

3 HYBRID RECOMMENDATION SYSTEM

A set of design principles arose as a result of these user studies. Recommendations were to be relevant, novel, diverse, and serendipitous. Additionally, users were interested in getting recommendations in the interface separated into different categories (e.g. "Sounds That Fit Your Tastes" and "Discover Different Kinds of Sounds"). Users were also interested in getting recommendations matching with semantic features of the sounds in their work-in-progress compositions (e.g. instrument, genre, key, and BPM).

The initial recommendation system we have developed does not yet support the entire set of user requirements that were illuminated by the user studies. It does combine collaborative filtering (using a statistical analysis of sound usage in past user scripts) and content-based filtering (using extracted audio features) to increase the relevance and novelty of the generated recommendations.

Recommendations are generated as follows:

(1) The algorithm takes in one or more sounds as its input. This input is the set of sounds that are already a part of a user's work-in-progress script/composition.
(2) The algorithm then generates a first list of sounds from the EarSketch sound library (the *co-usage list*) that have commonly been used in the past with the input sounds in scripts by any user.
(3) The algorithm then uses audio features of the sounds in the co-usage list to create a second list containing other sounds in the sound library that are acoustically similar to the sounds in the co-usage list (the *similarity list*).
(4) The algorithm removes sounds from the similarity list that have been commonly used with sounds in the co-usage list.
(5) Finally, the algorithm chooses sounds from the similarity list to present to the user as recommendations.

The co-usage list is an example of collaborative filtering (see the collaborative filtering section) and adds relevance to the generated recommendations by ensuring that recommendations are compatible with the set of sounds in the user's work-in-progress script/composition. The usage of the similarity list (rather than just the co-usage list) is an example of content-based filtering (see the content filtering section). The removal of sounds from the similarity list (that are commonly used with the co-usage list) adds novelty to the recommendations. The approach described here attempts to address diversity and serendipity of the generated recommendations, but explicit measures to ensure and evaluate these qualities is planned for future work (see future work).

3.1 Collaborative Filtering

The input to the collaborative filtering is the collection of sounds already being used in an active script at the time of recommendation generation. We take an *item-based approach* involving only an analysis of previous co-usage between sounds [13]. We take this approach to impose minimal collection of user information, such as user demographics and profile usage history, protecting EarSketch's primarily school-aged user base and conforming with its privacy policy [5]. The system returns a *co-usage list* of sounds in order of co-usage frequency. This co-usage is calculated using a sample set of 20,000 user scripts. Any sounds that are also in the input list are excluded to ensure that commonly co-used input sounds do not simply recommend each other.

3.2 Content-based Filtering

We compare two audio features to find sounds acoustically similar to the items in the *co-usage list*. These recommendations are the final output of the system. Recommended sounds are chosen based on their similarity to the most commonly co-used sounds comparing two properties of the audio signal — Short-Time Fourier Transform features and Mel-Frequency Cepstral Coefficients. The sounds are compared using the euclidean distance between their feature vectors, taken from the first 2 seconds of 48000 sample rate audio with a 1024-point Hann window and normalized for tempo.

Short-Time Fourier Transform Features $\mathcal{D}_{STFT}$ is the euclidean distance between the spectral density of two sounds, calculated using the librosa STFT function [8]. This function allows us to evaluate time-based similarities between sounds, and recommend sounds with similar function in a rhythmic context.

Mel-Frequency Cepstral Coefficients $\mathcal{D}_{MFCC}$ is the euclidean distance between the short-term power spectrum of two sounds, using the librosa MFCC function [8] [10]. This compares sounds in terms of temporally-independent energy, and acts as genre or instrument groupings.

Both features have been chosen due to their common usage in music information retrieval [6].

3.3 Recommendation Algorithms

This design aims to generate recommendations of sounds that are serendipitous to the user by not having high co-usage, and relevant through acoustical similarity to sounds that do. Diversity in recommendations is possible by including a high number of co-used sounds of a variety of styles. The multiple stages of randomness in both models, while not guaranteeing novelty, allow for different recommendations to be generated for the same combinations of inputs.

$\mathcal{N}$ represents an arbitrary factor limiting the amount of results gathered at different steps in the algorithms, and will be empirically determined during evaluation. The value of each variable labeled $\mathcal{N}$ in the below sections can be manipulated separately. This includes the lengths of the list of final recommendations, the *co-usage list*, and the *similarity list*.

The initial prototype of the recommender system is designed for use in standalone offline applications in addition to integration with the main EarSketch browser. Two recommendation algorithms were developed: one for live, real-time recommendation calculations and the other for faster server-side calculations. The first model, the dynamic model, conducts all calculations offline using pre-computed audio features to generate a list of recommendations for any combination of sounds. The static model, intended for online use, combines pre-computed lists of recommendations for any individual sounds to generate a single recommendation list.

3.3.1 Dynamic. The most commonly used sounds in conjunction with any of the input sounds parsed from a user script are found collectively using the collaborative filtering paradigm in the collaborative filtering section. Each commonly co-used sound is then compared to all other sounds in the EarSketch library, and a recommendation score for each is generated as the following equation:

$$S = \mathcal{D}_{STFT}^{-1} + \mathcal{D}_{MFCC}^{-1} + \mathcal{U} \tag{1}$$

where $\mathcal{D}_{STFT}$ = normalized STFT euclidean distance, $\mathcal{D}_{MFCC}$ = normalized normalized STFT euclidean distance, and $\mathcal{U}$ = normalized co-usage.

Additionally, STFT and MFCC distance from the original input samples are added or subtracted from the final recommendation score. This to generate recommendations that are either acoustically similar or different from the ones already found in the user script at the time recommendation. The sounds with the highest $\mathcal{N}$ recommendation scores are stored and joined together in a single *similarity list*. A random selection of $\mathcal{N}$ recommendations is chosen

from the highest $\mathcal{N}$ normalized recommendation scores in the master list, with higher priority given to the highest recommendations through fitness proportionate selection [2].

Static. The static model differs from the dynamic model in that it uses a pre-computed list of *similarity lists* generated for each individual sound in EarSketch, in order to make the recommendation algorithm less computationally intensive for server-side deployment. The lists for any combination of input sounds are joined together into a master list, and any duplicate sounds have their recommendation scores added and balanced by a factor of the square root of the number of lists. This method of balancing is in order to assign higher value to the strongest recommendations without drowning out the others, and is another scalable parameter that will be evaluated in future work (see the future work section). A random selection of $\mathcal{N}$ recommendations is chosen with higher priority given to the highest recommendations as with the dynamic model.

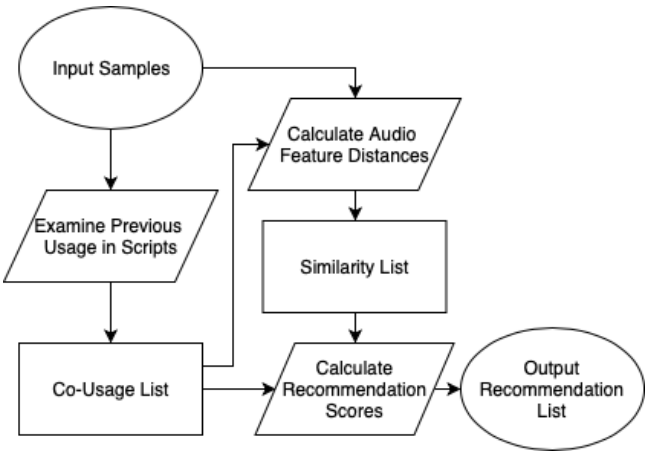

Figure 4: Program flow of the Dynamic recommendation system model, following the analysis of input samples to generate co-usage, similarity, and final recommendation lists.

4 FUTURE WORK

This algorithm is an exploratory stage of development and we plan to expand it along with the interface design with respect to current limitations information gained from user testing.

4.1 Recommendation System

The recommendation generation process will be modified to improve how it explicitly addresses its goals of relevance, novelty, diversity, and serendipity. Recommendation relevance will be improved by adding semantic metadata tags to the sounds, like instrument, genre, key, and BPM, and using those parameters (in addition to co-usage statistics and feature similarity) to select sounds. Novelty will be explicitly optimized for by measuring the distance between sounds in the lists and ensuring that recommendations are intentionally selected to be different from previously generated recommendations by some threshold novelty value **N**. Additionally, the calculations between audio features will be performed with operations and statistical measures other than euclidean distance, and will incorporate higher-level features such as rhythm. Similarly for a threshold diversity value **D**, recommendations would be chosen by adding sounds to a candidate set such that each new addition is at least **D** distance from every other item already in the set. Finally, serendipity will be explicitly optimized for by first collecting data searching for recommendations that are relevant but with low co-usage frequencies (indicating that they are rarely used together). Finally, each of the four recommendation generation goals will be weighted in order to tailor recommendations to different situations or different recommendation folders.

4.2 Proposed Evaluation

4.2.1 Recommendation System. Participants in a user study will empirically refine the various iterations of the recommendation system using different output-limiting values of $\mathcal{N}$, and different relative weighting of $\mathcal{D}_{MFCC}$ and $\mathcal{D}_{STFT}$. Additionally, they will be asked to choose sounds from the recommendation system and rate them in terms of relevance, novelty, diversity, and serendipity [13] for a combination of input sounds. The sounds they choose will be represented by the recommendation scores generated by each system iteration, in order to evaluate the weightings independently. Additionally, qualitative questions will reveal user opinions on other design aspects, like how many recommendations users want to see at once.

4.2.2 Interface Redesign. The current redesign has not been properly tested in a real world scenario, thus potential usability issues may arise with the navigation, language, and recommendation types. We will conduct moderated usability testing and record users' sessions interacting with a high-fidelity prototype while a researcher prompts them with tasks to complete. This testing will allow more information regarding EarSketch users' perceptions of a 'good' recommendation and how users will actually utilize these recommendations. As we move toward understanding how to recommend sounds to our user and better facilitate the exploration and discovery of sounds within EarSketch, our near-term goal is to iterate and improve on the proposed EarSketch redesign to accommodate recommendations.

REFERENCES

[1] Charu C Aggarwal et al. 2016. *Recommender systems.* Springer.

[2] Thomas Bäck. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, Inc., New York, NY, USA.

[3] S. Chang, A. Abdul, J. Chen, and H. Liao. 2018. A Personalized Music Recommendation System using Convolutional Neural Networks Approach. In *IEEE International Conference on Applied System Invention (ICASI).* IEEE, St. Petersburg Russia, 47–49. https://doi.org/10.1109/ICASI.2018.8394293

[4] Bram de Jong. 2005. Freesound. https://freesound.org

[5] Brian Magerko Jason Freeman. 2011. EarSketch. http://earsketch.gatech.edu/landing/

[6] Alexander Lerch. 2012. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics* (1st ed.). Wiley-IEEE Press.

[7] Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown. 2016. Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education (TOCE)* 16, 4 (2016), 14.

[8] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference.* 18–25.

[9] Tom McKlin, Brian Magerko, Taneisha Lee, Dana Wanzer, Doug Edwards, and Jason Freeman. 2018. Authenticity and Personal Creativity: How EarSketch Affects Student Persistence. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education.* ACM, 987–992.

[10] Paul Mermelstein. 1976. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence* 116 (1976), 374–388.

[11] Sergio Oramas, V.C. Ostuni, T. Di Noia, Xavier Serra, and E. Di Sciascio. 2016. Sound and Music Recommendation with Knowledge Graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8 (10/2016 2016), 1–21. https://doi.org/10.1145/2926718

[12] Gerard Roma and Xavier Serra. 2015. Music performance by discovering community loops. In *Proceedings of the Web Audio Conference (WAC), Paris.*

[13] E. Shakirova. 2017. Collaborative Filtering for Music Recommender System. In *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).* IEEE, St. Petersburg Russia, 548–550. https://doi.org/10.1109/EIConRus.2017.7910613

[14] Kai Siedenburg and Daniel Müllensiefen. 2007. Modeling Timbre Similarity of Short Music Clips. *Frontiers in psychology* 8, 1 (April 2007), 36–44. https://doi.org/10.3389/fpsyg.2017.00639